What is claimed is:

1. A method of providing a Certificate Status Service ("CSS") for checking validities of authentication certificates issued by respective issuing Certification Authorities ("CAs"), comprising the steps of:

identifying information needed for retrieving a status of an authentication certificate from an issuing CA that issued the authentication certificate;

configuring a connector based on the identified information for communicating with the issuing CA;

communicating with the issuing CA according to the configured connector when the status of the authentication certificate is queried; and

retrieving the status of the authentication certificate;

wherein the issuing CA and the connector are designated on a list of approved CAs in a configuration store.

2. The method of claim 1, wherein a local date and time are checked for whether they fall within a validity period indicated in the authentication certificate.

3. The method of claim 1, wherein the issuing CA is included in the list of approved CAs by vetting and approving the issuing CA according to predetermined business rules, and if the issuing CA is vetted and not approved, the issuing CA is designated on a list of not-approved CAs in the configuration store.

4. The method of claim 3, wherein vetting and approving the issuing CA includes registering a representation of a trusted authentication certificate with the CSS and adding at least the representation, status and a time-to-live data element to a local cache memory, and a connector is configured for retrieving the added status when the status of the trusted authentication certificate is queried.

5. The method of claim 2, further comprising the steps of checking a local cache memory for the status, and if the status is found in the local cache memory and the local date and time are within the validity period, retrieving the status from the local cache memory, wherein if the status is not found in the local cache memory or if the local date and time are not within the validity period, the CSS establishes a communication session with a certificate status reporting component of the issuing CA, composes a certificate status request according to the configured connector, retrieves the status from the certificate status reporting component, closes the communication

session with certificate status reporting component, and adds at least the authentication certificate's identification, status, and time-to-live to the local cache memory.

6. The method of claim 1, wherein the certificate status is indicated by a Certificate Revocation List (CRL), according to a publication schedule of the issuing CA, the CSS retrieves the CRL from a certificate status reporting component listed in the configuration store, the CSS clears a cache memory associated with the issuing CA, and the CSS determines the status of the authentication certificate from the CRL and stores the status in the cache memory associated with the issuing CA.

7. The method of claim 1, wherein the certificate status is indicated by a Delta Certificate Revocation List ("ΔCRL"); upon notification by the issuing CA that a ΔCRL is available, the CSS retrieves the ΔCRL from a certificate status reporting component listed in the configuration store; if the ΔCRL is a complete CRL, then the CSS clears a cache memory associated with the issuing CA, determines the status from the CRL, and stores the status in the cache memory; and if the ΔCRL contains only changes occurring after publication of a full CRL, the CSS determines the status from the ΔCRL, and stores the status in the cache memory.

8. The method of claim 1, wherein the communicating step includes communicating according to a sequence of connectors.

9. The method of claim 1, wherein a connector embeds more than one certificate status check in a single communicating step.

10. The method of claim 1, wherein the authentication certificate is not used for identification.

11. A method of retrieving a status of an authentication certificate issued by an issuing Certification Authority ("CA") in response to a query from a Trusted Custodial Utility ("TCU") to a Certificate Status Service ("CSS") to validate the authentication certificate's status, comprising the steps of:

locating and reporting the status if the status is present and current in a cache memory of the CSS;

otherwise performing the steps of:

obtaining a status type and retrieval method from a CSS configuration store;

if the status type is Certificate Revocation List ("CRL") and the status is not found in the cache memory, then reporting the status as valid;

if the status type is not CRL, then composing a certificate status request according to the status type;

establishing a communication session with the issuing CA;

retrieving the status from a status reporting component of the issuing CA using the obtained retrieval method and ending the communication session;

interpreting the retrieved status;

associating, with the interpreted retrieved status, a time-to-live value representing a period specified by a CSS policy for the status type;

adding at least the authentication certificate's identification, status, and time-to-live values to the cache memory; and

reporting the status to the TCU in response to the query.

12. The method of claim 11, wherein the CSS uses a certificate status protocol in the communication session.

13. The method of claim 11, wherein more than one status is retrieved using the obtained retrieval method.

14. The method of claim 11, wherein the authentication certificate is not used for identification.

15. A Certificate Status Service ("CSS") for providing accurate and timely status indications of authentication certificates issued by issuing Certification Authorities ("CAs"), comprising:

providing a status of an authentication certificate as indicated by a Certificate Revocation List ("CRL") when the certificate's issuing CA uses CRLs for indicating status;

otherwise, providing the status indicated by a cache memory when the cache memory includes a status and a time-to-live data element is not exceeded;

if the time-to-live data element is exceeded, clearing the status from the cache memory;

requesting and retrieving the status using a real-time certificate status reporting protocol when the status is not in the cache memory;

adding at least the certificate's identification, status, and time-to-live data element to the cache memory; and

providing the retrieved status.

16. The CSS of claim 15, wherein a status use-counter data element is added to the cache memory; the status use-counter data element is incremented or decremented every time the certificate's status is checked; and if the status use-counter data element passes a threshold, then the status is provided and the cache memory is cleared with respect to the status.

17. The CSS of claim 16, wherein a status last-accessed data element is added to the cache memory, and the status last-accessed data element in conjunction with the status use-counter data element enable determination of an activity level of the certificate's status.

18. The CSS of claim 17, wherein when a request is made to the CSS to retrieve a status of a new certificate and the cache memory has reached an allocated buffer size limit, the CSS searches the cache memory for a lasted-accessed data element indicating an oldest date and clears the respective cache memory entry; and the CSS then retrieves the requested status, places it in the cache memory, and provides the requested status.

19. A method of executing a transaction between a first party and a second party by transferring control of an authenticated information object having a verifiable evidence trail, comprising the steps of:

retrieving an authenticated information object from a trusted repository, wherein the authenticated information object includes a first digital signature block comprising a digital signature of a submitting party and a first authentication certificate relating at least an identity and a cryptographic key to the submitting party, a date and time indicator, and a second digital signature block comprising a second digital signature of the trusted repository and a second authentication certificate relating at least an identity and a cryptographic key to the trusted repository; the first digital signature block was validated by the trusted repository; and the authenticated information object is stored as an electronic original information object under the control of the trusted repository;

executing the retrieved authenticated information object by the second party by including in the retrieved authenticated information object a third digital signature block comprising at least a third digital signature and a third authentication certificate of the second party; and

forwarding the executed retrieved authenticated information object to a trusted custodial utility ("TCU"), wherein the TCU verifies digital signatures and validates

authentication certificates associated with the digital signatures included in information objects by at least retrieving status of the authentication certificates from a Certificate Status Service ("CSS") provided according to claim 1; the TCU rejects a digital signature block if the respective digital signature is not verified or the status of the respective authentication certificate is expired or is revoked; and if at least one signature block in the information object is not rejected, the TCU appends the TCU's digital signature block and a date and time indicator to the information object and takes control of the object on behalf of the first party.

20. The method of claim 19, wherein a signature block includes at least one hash of at least a portion of the information object in which the signature block is included, the at least one hash is encrypted by the cryptographic key of the block's respective signer, thereby forming the signer's digital signature, and the signer's digital signature is included in the signature block with the signer's authentication certificate.

21. The method of claim 20, wherein the executing step includes displaying a local date and time to the second party, affirming, by the second party, that the displayed local date and time are correct, and correcting the local date and time if either is incorrect.

22. The method of claim 19, wherein if the TCU rejects a digital signature block, the TCU requests a remedy that requires the digital signature to be recomputed and the signature block to be reforwarded.

23. The method of claim 19, wherein the TCU checks the local date and time for accuracy and that they are within a validity period indicated by the second party's authentication certificate.

24. The method of claim 23, wherein if the local date and time are not within the validity period indicated by the second party's authentication certificate, the TCU notifies the second party that the authentication certificate is rejected and the first party that the transaction is incomplete.

25. The method of claim 19, wherein one or more digitized handwritten signatures are included in the information object, and placement of the digitized handwritten signatures in a data structure is specified by at least one signature tag.

26. The method of claim 19, wherein placement of one or more signature blocks in a data structure is specified by at least one signature tag.

27. The method of claim 26, wherein one or more signature blocks are separately forwarded to the TCU with respective signature tags, and the TCU validates the signature blocks by:

rejecting a signature block if either the respective digital signature is not verified or the respective authentication certificate is not validated, and

placing the signature block according to the respective signature tag if the signature block is not rejected,

wherein, to signature blocks sent separately, the TCU adds a date and time indication to each signature block and appends according to business rules the TCU's signature block in a wrapper that encompasses the information object and placed signature blocks.

28. The method of claim 27, wherein the TCU verifies a digital signature and validates an authentication certificate in a signature block by:

determining from the business rules whether a party associated with the authentication certificate has authority,

verifying the party's digital signature,

checking that the authentication certificate's validity period overlaps the TCU's current date and time,

checking that the local date and time falls within an allowable deviation from the TCU's current date and time, and

retrieving status of the authentication certificate from the CSS, and

if any of the preceding steps results in an invalid or false output, the digital signature is deemed invalid, the transaction is not executed, otherwise the digital signature is deemed valid and the transaction is executed.

29. The method of claim 19, wherein the CSS provides authentication certificate status to the TCU by at least the steps of checking a local cache memory for the status, and if the status is found in the local cache memory and the local date and time are within the validity period, and retrieving the status from the local cache memory; if the status is not found in the local cache memory or if the local date and time are not within the validity period, the CSS establishes a communication session with a certificate status reporting component of the issuing CA, composes a certificate status request according to the configured connector, retrieves the status from the certificate status reporting component, closes the communication session with certificate status reporting

component, and adds at least the authentication certificate's identification, status, and a time-to-live data element to the local cache memory.

30. The method of claim 19, wherein the first party is a first TCU and the transaction is for transferring custody of one or more electronic originals to the first TCU from a second TCU, an owner of the transaction provides the second TCU with a manifest that identifies electronic originals to be transferred to the first TCU, the second TCU establishes communication with the first TCU and identifies the purpose of its actions, the manifest is communicated to the first TCU so that it is able to determine when the transfer of custody has been completed, the second TCU transfers each identified electronic original to the first TCU, the first TCU retrieves status of the second TCU's certificate and verifies the second TCU's digital signature on each transferred electronic original, if any of the second TCU's digital signatures or certificates are invalid, then the first TCU notifies the second TCU and seeks a remedy, if the second TCU does not provide a remedy, the first TCU notifies the transaction owner that the requested transfer of custody has failed, otherwise the second TCU creates a new wrapper for each successfully transferred information object, adding a date-time stamp and the first TCU's signature block.

31. The method of claim 30, wherein the transaction is a transfer of ownership in response to an instruction, transfer of ownership documentation is placed in either the first TCU or the second TCU, the TCU having the transfer of ownership documentation validates authenticity of the transfer of ownership documentation by verifying all digital signatures, certificate validity periods, and using the CSS to check certificate status of all authentication certificates included in the transfer of ownership documentation, appends a date and time indication, and digitally signs, wraps and stores the transfer of ownership documentation, which are added to the manifest.

32. The method of claim 19, wherein certificate status is indicated to the CSS by a Certificate Revocation List ("CRL"), according to a publication schedule of the issuing CA, the CSS retrieves the CRL from a certificate status reporting component listed in the configuration store, the CSS clears a cache memory associated with the issuing CA, and the CSS determines the status of the authentication certificate from the CRL and stores the status in the cache memory associated with the issuing CA.

33. The method of claim 19, wherein certificate status is indicated to the CSS by a Delta Certificate Revocation List ("ΔCRL"); upon notification by the issuing CA that a

ΔCRL is available, the CSS retrieves the ΔCRL from a certificate status reporting component listed in the configuration store; if the ΔCRL is a complete CRL, then the CSS clears a cache memory associated with the issuing CA, determines the status from the CRL, and stores the status in the cache memory; and if the ΔCRL contains only changes occurring after publication of a full CRL, the CSS determines the status from the ΔCRL, and stores the status in the cache memory.